On the applicability of requirements determination methods

Bollen, Petrus Wilhelmus Laurentius

CHAPTER 7

CONCLUSIONS, GENERAL DISCUSSION AND RECOMMENDATIONS

7.1 INTRODUCTION

In this final chapter of this thesis we will evaluate our main research question that we have stated in chapter 1:

"Does there exist a requirements determination method that is applicable in a wide range of business organizations and that can be used for specifying the complete domain requirements for a given business application subject area in an efficient, and formal way ?"

In chapter 1 we illustrated the relevance of the field of requirements determination. We have concluded that the theory development on the field of our research topic: *requirements determination*, has taken place in a number of fields of study. Among those fields of study are: *management information systems, information systems development methodologies, speech-act theory, ontology* and *conceptual modeling*. Furthermore, we concluded that the theory development on the field of requirements determination has been plentiful but a sound methodology for the specification of application requirements is missing. In chapter 1 we have also pointed at a MIS research niche that we want to exploit. This research niche is concerned with the *semantic verification* of (initial) requirements (Dullea et al., 2003). This has lead us to formulate the design object of the research in this study as: the development of a method for requirements determination that has modeling provisions that guide an analyst in elicitating the initial requirements from domain users and that contains a semantic verification or validation procedure that guarantees user validation of the requirements.

We have sketched a research approach that is suitable for our research purposes: the 'design-research' approach. In this approach we have applied the 'design-research'-cycle (Van Engelen and Van der Zwaan, 1994) in which we first have to establish a design objective and subsequently a design specification in order to be able to evaluate existing designs for a requirements determination method.

The design specification for a requirements determination method was given in chapter 2 in which we have synthesized from the literature, four (groups of) criteria that a requirements determination method must comply to: *domain richness, completeness, efficiency* and *formality*.

In chapter 3 we have covered the first group of 'alternative' designs, namely an evaluation of the existing alternative designs for requirements determination methods by surveying the existing literature on requirements determination, e.g. DFD's ISAC, EER, UML, ARIS and ORM. In the research methodology literature this is called the 'evaluation problem'. In chapter 3 we have concluded that no single approach fulfills the design specification that was given in chapter 2. In case no single existing design can be found that conforms to our design specification we need to develop an alternative design that must fulfill the requirements that we have postulated in the final research question (the development problem).

In chapter 4 of this thesis we subsequently have derived a detailed design specification that resulted in 19 requirements method demands (RMDs) for a 'to-be' designed RDM.

In chapter 5 we have documented the way of modeling for a proposed requirements determination method: Natural Language Modeling (NLM) and in chapter 6 we have documented the way of working and the way of controlling for NLM. Chapters 5 and 6, therefore, constitute the results of the generation of an alternative design as is given in the 'generation of alternative designs' stage from the 'design-research'-cycle (Van Engelen and Van der Zwaan, 1994).

We concluded in sections 5.10 and 6.10 that the NLM requirements specification language, its set of accompanying modeling algorithms and it's project management precedence logic jointly fulfill all 18 design requirements that were derived in chapter 4 for a to be designed RDM. Hence, we can conclude that Natural Language Modeling (NLM) is a requirements determination method (*RDM*) that complies to the design objective in this thesis. Natural Language Modeling, therefore, provides the answer to our main research question from chapter 1.

7.1.1 Organization of chapter 7

This chapter is organized as follows. In section 7.2 we will summarize the research findings for the research sub-questions that were derived in chapter 1. Furthermore, we will, show in section 7.2 how the answers to the sub-questions of our research lead to the answer to our definite research question. In section 7.3 we will defend our research methodology from a retrospective point of view. In section 7.4 we will give recommendations for future research areas in the field of Management Information Systems. In section 7.5 we will give recommendations for practitioners in the MIS field. Finally, in section 7.6 we will reflect upon the research questions for this thesis, our research approach, the research outcome and the research process.

7.2 RESEARCH FINDINGS

In chapter 1 we have introduced the subject of this study: requirements determination. We have also sketched the application of computerized information systems in the past 50 years and we have shown that in recent history the emphasis within the application of information systems in businesses has been on ERP systems. We concluded that a complete and consistent requirements specification is still needed as a starting point for the customisation and implementation of an ERP system. Existing approaches for requirements determination often use a natural language statement of the initial requirements as a starting point for the creation of a requirements specification (Goldin and Berry, 1997). However, none of the tools that were mentioned in this study give much help on how such an (initial) language statement can be obtained.

The application of the 'design-research'-cycle to carry out the research that will enable us to achieve the (preliminary) goal of our research will lead to a number of research sub-questions.

Research sub question 1 :

> What are according to the existing requirements determination literature, the quality criteria for a requirements determination method that can be used for eliciting, verifying and specifying the complete domain requirements for a given business application subject area in a wide range of business organizations in an efficient and formal way ?

Research findings for sub question 1 :

We have synthesized (four groups of) criteria for a requirements determination method: *domain richness*, *completeness*, *efficiency* and *formality.*

Domain richness criterion

A literature review of the requirements determination literature has lead to four dimensions for the domain richness criterion.

The first dimension that characterizes a domain is what we have labelled the dimension *perception*. The actual 'value' on this dimension for any given domain can range from "uniform for all users" (*similar* perception) to "different for all users" (every user has a *different* perception of a underlying reality) (Galliers and Swan, 2000).

The second dimension that characterizes a domain is labelled the dimension *turbulence.* This dimension actually represents the extent (or frequency) in which the rules, information and procedures in an application domain are subject to change (Land, 1998).

The third domain dimension that we have derived from our literature study in chapter 2 is the dimension *tacitness.* The tacitness can range from a fully 'tacit' application domain in which no single knowledge creating process is explicit to a fully 'explicit' UoD in which every knowledge or information generating process can be made explicit.

The fourth dimension for domain richness, is the dimension *anchoring* (Bubenko and Wangler, 1992; Flynn and Warhurst, 1994), ranging from a 'tangible' starting point for the requirements determination process to an 'abstract' starting point.

Completeness criterion

The completeness criterion for a requirements determination method has been operationalized along two dimensions that define *what* must be incorporated in a requirements specification for application domains. The first dimension is the perspective dimension: the *data-oriented* perspective, the *process-oriented* perspective and the *behaviour-oriented* perspective. The conceptual data-oriented perspective should concentrate on the business data and must capture the domain concepts, the definition and the naming conventions for those domain concepts, the semantic relationships between the domain concepts and other ‘static’ and ‘structural’ knowledge in the enterprise. The process-oriented perspective should be able to capture the business activity and user perceivable tasks and describe the ‘cross-reference’ on how the ‘elements’ in the static structure are created, or what procedures exist for the creation of instances of semantic relationships. Finally, the behaviour-oriented perspective describes how ‘events’ can be cross-referenced to ‘elements’ in the process- and data-oriented perspectives. This means that any requirements specification should potentially consist of models that covers these three (conceptual) perspectives. The second dimension is concerned with the question what elements must be contained in every perspective (see table 7.1)

Table 7.1 Types of rules within perspectives for completeness criterion

	state	**state**	**action**
Data-oriented	Data model	Static constraints	Dynamic constraints
Process-oriented		Static derivation	
Behaviour-oriented			Dynamic rules

Efficiency criterion

Another criterion that we can use for evaluating requirements determination methods is concerned with the amount of resources that are needed to create a requirements specification when such a requirements determination method is applied in a given application UoD. This criterion is generally known as *efficiency*. The operationalization of this criterion for the purpose of evaluating requirements determination methods has taken place for the *way of modeling* and the *way of working* as well as the *way of controlling*.

With respect to the *way of modeling* the number of equivalent modeling constructs in the specification language determines the value on this criterion,

With respect to the *way of working* of a requirements determination method we can say that the availability of a (set of) procedure(s) that guides an analyst in the requirements determination project will determine the efficiency of the way of modeling of requirements determination method.

With respect to the *way of controlling* we can define efficiency on two areas. Firstly, the area of quality management. In this philosophy, quality deficiencies must be prevented by having a number of ‘quality-checking’ sub-procedures. Secondly, the way of controlling is concerned with the project management of the requirements determination project. The efficiency regarding these project management issues must be measured in terms of three project targets: *performance, cost* and *time* (Mantel et al., 2001)

Formality criterion

The relevant formality dimension to which a requirements specification must comply are the following: *consistency* and *preciseness*. This means that the modeling constructs that are used for creating requirements specifications in the different perspectives must be formally defined, in order to prove their consistency. Secondly, the way of working, must be formal: a formal modeling procedure(s) must exist that precisely specifies how the consistent modeling constructs that were defined in the way of modeling, must be instantiated in a requirements determination project in order to obtain semantic correctness in complicated application subject areas.

With respect to the way of controlling we must be able to formalize the planning of activities that have to be carried out in a requirements determination project, for example in a precedence diagram and we must be able to give provisions that enable traceability.

Research sub question 2 :

> Why do the existing requirements determination approaches from the literature not comply with the quality criteria for assessing requirements determination methods ?

Research findings for sub question 2 :

Research findings for the domain richness criterion

The application of a requirements determination method must lead to a requirements specification that reflects the (possibly) different perceptions of an underlying reality by different user groups. It is possible to reflect these different perceptions by using the EER, UML and ORM approaches, whenever they are embedded in a procedure that enables an analyst to integrate the different views from different user groups on the 'underlying reality' by integrating the sub-schemas of these users into a final 'overall' requirements specification in which the different perceptions are made explicit. The EER, UML and ORM approaches that we have discussed in chapter 3 do not give provisions for this.

The 'turbulence' dimension characterizes the extent in which an application domain is subject to changes in the business data and business rules. We concluded that the EER and UML approaches are most prone to remodeling because of the multitude of information bearing constructs. ORM has a problem with a multitude of naming conventions which might lead to unstable models.

With respect to the 'tacitness' dimension, the EER, UML and ORM approaches basically have the assumption that users will be able to express their initial requirements in natural language. This restricts the applicability of these approaches to those domains that exclusively contain explicit knowledge

With respect to the 'anchoring' dimension, the requirements determination processes in which we use EER and UML models for our specification language are in principle not limited to any specific range on the anchoring scale. ORM is anchored in *familiar examples* and it requires the domain expert to come up with these real examples and therefore is applicable for those domains that are on the 'tangible' side of the anchoring scale. The initial language in ORM is the language of verbalizable *familiar examples* and it requires the domain expert to verbalize these examples in (a subset of) natural language. The initial language in EER and UML is not specified but it can be anything because no procedure is given how to get from an initial requirements statement to the EER diagram or UML model(s).

Research findings for the completeness criterion

With respect to the encoding capabilities of a given approach for the data model, the static constraints, the dynamic constraints, static derivation rules and dynamic rules, the main conclusion is that no single approach is able to comply fully with the *completeness* criterion. There exists a large difference between the families of approaches and even between members within a given family in terms of the extent in which the application domain semantics can be expressed in the *data model,* and as *static* or *dynamic constraints*, *static derivation (rules)* and *dynamic rules*. Furthermore, the existing approaches, generally, lack a formalized way of working that will assure completeness, in the sense that all existing relationships and constraints in an application subject area, will be 'detected' by the analyst in the requirements determination project. This means that there still is an opportunity to improve the requirements determination approaches we have surveyed in chapter 3 in terms of completeness.

Research findings for the efficiency criterion

With respect to the *efficiency* criterion we must remark that in EER and UML in a number of cases remodelling is necessary because of the application of the attribute modeling construct in the initial requirements specification. The main finding of this literature survey is that 2 out of these 3 approaches use more than 2 information bearing constructs which can lead to instable requirements specifications. Furthermore, the non-existence of a precise modeling procedure in all approaches might lead to unnecessary rework in the requirements determination process, because verification is not enforced. Furthermore, with respect to the way of controlling, the existing approaches do not cover the project management and quality assurance steps.

Research findings for the formality criterion

With respect to the *consistency* dimension we can conclude that in many (E)ER approaches and in the UML it is not possible to use a single definition for minimum cardinalities or multiplicities across all types of semantic relationships. In UML it is not clear how the modeling concepts that are used in the 9 different diagram types are related on the level of an application requirements specification. In ORM an inconsistency is found with respect to the treatment of derived fact types, sometimes they will be contained in the application information grammar sometimes they will not.

With respect to the *preciseness* dimension we remark that the optionality of some modelling constructs in all three approaches that we've studied might lead to imprecise requirements specifications. With respect to the questioning of assumptions we can conclude that in EER, UML and ORM no procedure exists that allows an analyst to question the assumptions on which the utterance of the domain semantics is based. The position of these approaches is basically that the domain requirements that are uttered by the user are encoded in the model 1-on-1. ORM claims to perform checks on sample populations, however, it does not give guidelines on how to formally perform these checks in a dialogue with the responsible domain user.

Table 7.2 Requirements method demands for the way of modeling

RMD	Requirements method demands for the way of modeling
1	A to-be designed RDM must contain 1 information bearing modeling construct. This construct must be able to express the complete, precise and consistent communication semantics of any N-ary relationship
2	The modeling construct(s) for naming conventions must allow for one domain-based naming convention and must be able to capture the semantics regarding the context in which the naming convention is valid.
3	The to be designed requirements method must contain a role construct and an explicit naming convention for roles
4	The static constraint types in the to-be designed requirements method must at least contain those types that enable us to encode those business rules that can be encoded by relationship cardinalities in EER and UML
5	A requirements specification that is the result of the application of the to-be designed requirements determination method must be able to adapt to an evolving application logic without unnecessary remodeling
6	The definition of an application object or entity in the to be designed requirements method must not imply that it can exist on its own by default
7	The definition of the static constraint types in the to-be designed requirements method must be the same for all arities of the semantic relationships in the data model and must contain an explicit reference to the elements in the data model.
8	The definition of the dynamic constraint types in the to-be designed requirements method must enable us to explicitly refer to the (actual and projected states of the) application's data base
9	The definition of static derivation (rules) in the to-be designed requirements method must contain an explicit reference to the elements in the data model that serve as an input for the static derivation (rule) and it must contain a precise specification on how these elements lead to the result of the static derivation (rule)
10	An internal event in the to-be designed requirements method must be defined as the insertion or deletion of a specific piece of domain knowledge into or from the application's data base An external event in the to-be designed requirements method must be defined as something that happens in the application domain and that can lead to the insertion or deletion of a specific piece of domain knowledge into or from the application's data base or the execution of a static derivation rule (eventually) under some condition on the content of the application's data base
11	A condition in the to-be designed requirements method must be defined as a proposition on the application's information base that must yield the value true or false when evaluated at any point in time

In chapter 3 of this thesis it was concluded that for the three requirements determination approaches that were studied in detail in this chapter (E)ER, UML and ORM no single approach fulfills all the quality criteria for a RDM that were derived in chapter 2.

Research sub question 3 :

> What are the necessary elements for the way of modeling, the way of working and the way of controlling for a requirements determination method so that this method complies with the quality criteria that we have given for the design specification ?

Research findings for sub question 3 :

The diagnosis of these modeling deficiencies in the state-of-the-art in requirements determination has lead to the formulation of 18 requirement method demands (RMD's) for the specification of a to-be designed requirements method in chapter 4. We have divided the 19 RMD's into RMD's for the way of modeling (table 7.2), RMD's for the way of working (table 7.3) and RMD's for the way of controlling (table 7.4).

Table 7.3 Requirements method demands for the way of working

RMD	Requirements method demands for the way of working
12	The definition of the modeling constructs for the data model in the to-be designed requirements method must be accompanied by some kind of guidance on how all instances of these modelling constructs can be found in an application subject area. The definition of the constraint types in the to-be designed requirements method must be accompanied by some kind of guidance on how such instances of a constraint type can be found in an application subject area.
13	A view integration sub-procedure must be defined in the to-be designed RDM in which it is specified how an analyst must carry out the integration of views on the application domain by user (groups) that have a different perception on the 'underlying' reality
14	The to-be designed requirements determination method must provide facilities for transforming implicit tacit knowledge into explicit knowledge
15	The to be designed requirements determination method must accommodate every possible starting point in the requirements determination process ranging from abstract to tangible; ranging from natural language description to documents that can only be understood by domain users.
16	Formal modeling procedure(s) must be defined in the to-be designed requirements method in which it is precisely specified how an analyst must carry out a modeling step in the most efficient way.

Table 7.4 Requirements method demands for the way of controling

RMD	Requirements method demands for the way of controlling
17	The way of working in the to-be designed RDM must have explicit formal quality assuring sub-procedures for the activities in the work breakdown structure and formal checks that enables an analyst to validate the information that is supplied by the user and that confronts a domain user with his/her assumptions and enables a user to validate the information that is supplied to the analyst
18	The way of working in the to-be designed requirements determination method must have a work breakdown structure that allows to formally plan the activites in a requirements determination project.
19	The way of modeling and the way of working in the to-be designed RDM must have provisions that enable traceability.

We now go back to our main research question

Main research question :

> Does there exist a requirements determination method that is applicable in a wide range of business organizations and that can be used for specifying the complete domain requirements for a given business application subject area in an efficient and formal way ?

Research findings for main research question :

In chapter 5 NLM's way of modeling was defined. The modeling constructs for the *specification* of an application requirement in a basic information model and the accompanying constraints and their naming conventions were given. Furthermore, their applicability and generalizability in business UoD's was illustrated. In the first part of chapter 6, the elements for the way of working in NLM were defined, consisting of procedures or algorithms that specify how an analyst must carry out the requirements *elicitation* process in a dialogue with a knowledgeable domain user. Every procedure or algorithm contains built-in quality preserving and verification step(s) that verifies the recorded requirements segment (generally) in a dialogue with the domain user. In the second part of chapter 6 the elements in the way of controlling for NLM, were given in which the (project) management of the requirements determination process using the NLM method was illustrated.

In chapters 5 and 6 we have defined a requirements determination method that contains the necessary elements as they were laid down in 19 RMD's from chapter 4 and which therefore gives an answer to our main research question from chapter 1. The Natural Language Modeling (NLM) requirements determination method turns out to fulfill all necessary requirements for a to-be requirements determination method as was defined in chapter 1. In figure 7.1 we have shown how the research (sub)-questions are related.

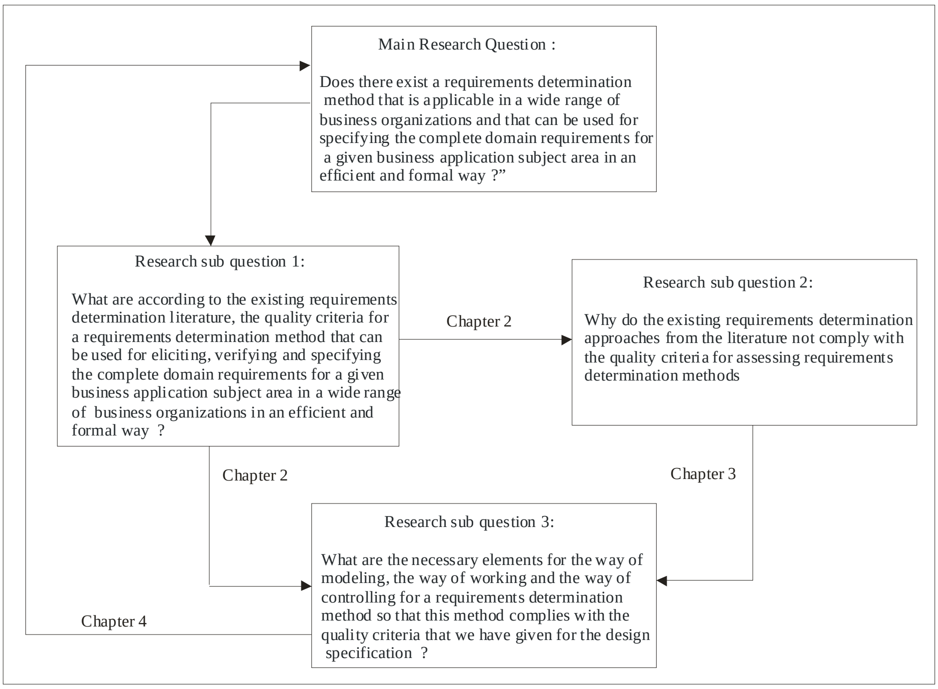

Fig. 7.1 Relationship between research (sub) questions)

The main findings regarding the extent in which NLM satisfices the 19 demands that we have derived are summarized here.

Findings for the way of modeling

The NLM requirements specification language contains only one information bearing construct: the fact type and it allows us to model any naming convention and semantic connection. The introduction of the sentence group template construct and the application concept repository allows us to capture the complete domain semantics of the UoD and therefore fulfills requirement *RMD 1*. The introduction of naming convention fact types and compound referencing schemes in combination with an accompanying sentence group template that enables us to capture the context in which the naming convention is valid fulfills requirement *RMD 2*. The definition and consistent application of the role construct and the mandatory naming convention from such a construct within the UoD of an analyst in the NLM specification language fulfills requirement *RMD 3*. Furthermore we have given modeling provisions that allows us to define any type of static constraint that currently exists within the EER and UML (compliance to *RMD 4*). It was also shown that NLM leads to requirements specifications that can easily evolve with changing application requirements *(RMD 5*). If instances of an intention can only exist on their own, this can be modeled as a unary fact type. This means that the NLM modeling constructs comply with requirement *RMD 6*. The definition of the uniqueness- and set-comparison constraints is fully scalable as a function of the arity (N). This means that we have complied with requirement *RMD 7*. With respect to the transition constraints we remark that in our legend we have explicitly incorporated the relationship that the constraint has in terms of the values of the roles that are involved and it contains explicit references to before- and after- states of the application information base and therefore, NLM complies to *RMD 8*. The derivation rule constraints contain a reference to the roles from the Basic information Model of the UoD. This means that the derivation rule constraint that we have defined in the NLM's way of modeling complies with *RMD* 9. We have made a distinction into internal and external events in NLM. This leads to the compliance to *RMD 10*. In the impulse, an information base condition can be contained. Such an information base condition (IBC) is evaluated at some point in time. If the application information base at that point in time in combination with the information base condition yields the value true than the derivation rule and/or insert/delete operation will be executed. If it yields the value false nothing will happen. This means that requirement *RMD 11* has been fulfilled.

Findings for the way of working

The application of algorithms 1 through 4 will lead to the detection of all semantic relationships and naming conventions in the application subject area. We have specified static constraint derivation algorithms 6 and 7 to detect all uniqueness and set comparison constraints. In order to derive all instances of the dynamics constraints we have specified transition constraint derivation algorithm 8 as sub-procedure in NLM's way of working. In order to derive all instances of the derivation rule constraints we have specified the derivation rule constraint algorithm 9 in which the precise specification (or derivation formula) can be established. In algorithm 10 we have incorporated the question in which an internal event can lead to the execution of a derivation rule or another information base event. Furthermore, the algorithm systematically confronts the users in the SoI with derivation rules and tries to elicit the potential 'external' events that might invoke such a derivation rule. We can conclude that *RMD 12* has been fulfilled. In algorithm 5 a view integration algorithm has been defined. This fulfills *RMD 13*. The application of the natural language axiom in an organizational setting in which domain users are enabled to make implicit knowledge, explicit fulfills requirements *RMD 14*. The application of the natural language axiom in NLM also allows us to apply NLM in many organizational settings, ranging from abstract to tangible UoD's and from natural language descriptions to other descriptions that can only be understood by users. This leads to compliance to *RMD 15*. The sub-division of the modeling procedures in NLM's way of working into a number of formal algorithms has been done in such a way that the amount of analysis steps that have to be performed by (an) analyst(s) is minimized and therefore NLM fulfills *RMD 16*.
The precise specification of the NLM modeling procedure in a number of algorithms with built-informal quality assurance checks fulfills requirements *RMD 17*.

Findings for the way of controlling

The way of working in NLM has a work breakdown structure that consists of 10 activities or transformations that are laid down as formal algorithms and therefore can be formally planned as activities in a requirements determination project according to

RMD 18. Furthermore, NLM contains provisions that enable traceability in the requirements determination processes, by forcing an analyst to use naming conventions for the concept that he/she uses in the process of requirements determination and therefore NLM fulfills requirement *RMD 19*.

7.3 RESEARCH METHODOLOGY

In sections 5.10 and 6.10 we have already concluded that the NLM requirements determination method complied with all 19 requirements method demands RMD's that were specified in chapter 4. In this concluding chapter of this thesis we will reflect on the final stage from the 'design-research'-cycle: selection of the desired design from the set of alternative designs.

We can now conclude that NLM is a 'satisficing' solution to our main problem statement from chapter 1, since it 'satisfices' all 19 demands for a requirements determination method that were derived, based upon the literature research on the state-of-the-art in requirements determination methods. The research goal that was phrased in chapter 1: "to develop a method for requirements determination for which the way of modeling allows the analyst to capture all business entities and all business rules. This to-be developed RDM should have a way of working that contains modeling provisions that guide an analyst in elicitating the initial requirements from domain users. Finally, this method's way of controlling must contain quality preserving procedures that guarantees that a requirements specification that is the result of the application of this method have been validated by the user(s)" .., therefore, is achieved, by developing an alternative design that complies to the requirements that were derived in chapter 4.

7.4 FUTURE MIS RESEARCH PROPOSALS

We will conclude this chapter with a number of topics for future research.

We have documented the NLM requirements determination method in chapters 5 and 6 of this thesis. The NLM requirements determination method clearly, provides a number of advantages over the 'state-of-the-art' in requirements determination methods. One of the most distinguishing features of NLM compared to for example, (E)ER, ORM or UML is the way in which the application of modeling constructs, for example, constraint types, is made explicit in a 'constraint'-legend and an accompanying 'instantiation algorithm'. In the appendix A to this thesis, we have provided the readers with some example constraint legends and in chapter 6 we have shown the accompanying instantiation algorithms for these constraint types. An agenda for future research, is to define more (in the sense of 'orthogonal' to the existing constraint types in this thesis) constraint types, that prove to be significant for business application subject areas, but above all, to develop accompanying instantiation algorithms that can be used in an analyst-domain user dialogue, and that are based upon the acceptance and/or rejection of ‘real-life’ examples by knowledgeable domain users.

7.5 RECOMMENDATIONS FOR PRACTITIONERS IN THE MIS FIELD

We can conclude that the NLM requirements determination procedure explicitly shows the separation of concerns between the analyst and the user in the process of requirements modeling by providing the *semantic bridges* for this analyst-user dialogue. In addition to the creation of a NLM requirements specification that is an allowed extension of the NLM meta model we need guidance on what specific extension of this meta model reflects the domain semantics in a *precise* and a *complete* way. We have shown that such a semantic correct specification will be achieved when the algorithms that we have introduced in this thesis will be applied in a requirements determination project in which the sequence of their application is performed under the precedence requirements that were given in chapter 6. Furthermore, this will result in the most *efficient* way of working. Although a number of procedures in the NLM requirements determination method at first sight have a ‘trivial’ appearance, the consistent application of the procedures in the ‘way-of-working’ in this thesis in practice has proven to improve the ‘quality’ levels of the resulting information models, because even the experienced analyst can always ‘fall back’ on the procedure in those situations in which the application subject area becomes too complex. Another advantage is that inexperienced analysts will be able to create requirements specifications that have the same quality level as the specifications that are created by experienced analysts. In a project in which the NLM requirements determination method is applied for the creation of a requirements specification, the division into sub-projects and the order in which these sub-projects are executed does not have an impact on the final specification.

As we pointed out earlier, the objective of this thesis research was to develop a requirements modeling language and a modeling procedure, rather than to specify an ‘optimal’ notation legend for such a language. Practitioners, however, need to be able to communicate, with domain users, management and peers in many cases using a pre-defined diagramming technique or notational legend. The conclusions from this research in terms of giving a preference to a modeling language that has *one* information bearing construct, that has uniform modeling facility for naming conventions and that has a number of *orthogonal constraint types.* The definition of these constraint types in combination with an instantiation procedure has a built-in guarantee that these constraint instances can always be derived in any application UoD, whenever the appropriate instantiation procedure is applied in combination with a knowledgeable domain user. The practitioner should evaluate the requirements specification/determination approach that he/she is currently using. After this evaluation the practitioners can decide to limit or redefine the modeling constructs that they want to keep and define new modeling constructs if one more necessary constructs are missing. In a second stage a ‘notational’ legend must be (re)defined that preferably is ‘backwards compatible’ with the old way of modeling and the old way of working. In the third stage of the evaluation of the current approach, practitioners must decide on what types of constraints are relevant for the specific type(s) of application domain(s) in which the requirements determination method is going to be applied. We emphasize that for these constraint types, accompanying procedures must be specified on such a level of concreteness that an analyst can apply these procedures in a dialogue with a knowledgeable user.

7.5.1 Application of NLM in practice

The NLM requirements determination that we have documented in chapters 5 and 6 has been applied by master students in MIS a number of times in large and small enterprises, see for example Bogget (1994) and Enter (1999). Other students have applied this approach on object-oriented models (Clayes, 1996) and on the accounting knowledge domain (Wolthuis, 1997).

7.6 CONCLUDING REMARKS

In this thesis we have studied the field of requirements determination for enterprise information systems. We discovered that in the ERP era that characterizes the information systems in many (large) enterprises at the beginning of the 21st century, the issue of requirements determination is still relevant. We also discovered that the ‘state-of-the-art’ of this field still shows a number of omissions in the definition of requirements specifications modeling constructs and methodology. We have chosen to specify ‘the requirements’ or demands for a requirements method itself using four (groups of) criteria. These criteria were operationalized by studying the three most dominant requirements determination approaches that exist today. These criteria were subsequently translated into 19 specific demands (RMD’s) for a to be designed requirements determination method. In the second part of this thesis (chapters 5 and 6) we have introduced the Natural Language Modeling (NLM) approach for requirements determination. It turns out that NLM satisfices all 19 requirements and therefore can be considered an appropriate design alternative, as is specified in the design research cycle (Van Engelen and Van der Zwaan, 1994). Hence the choice of Natural Language Modeling as answer for our research objective is justified.

7.7 REFERENCES

Bogget, M. (1994): Implementation of a Management Reporting System & Preparation of MRP model at PTZ Nelahiozeves Unilever. Final Thesis Business Economics. University of Maastricht

Bubenko, J., Wangler, B. (1992): Research Directions in conceptual specification developments. In: Conceptual Modeling, Databases, and Case (Loucopoulos, P., Zicari, R. (eds.)). Wiley. 389-412

Clayes, C. (1996): Final thesis. International business studies. University of Maastricht.

Dullea, J., Song, I-Y., Lamprou, I. (2003): An analysis of structural validity in entity-relationship modeling. Data & Knowledge Engineering **47**: 167-205

Enter, N. (1999): The semantics of the CIC SAP R/3 core. Final thesis. International business studies. University of Maastricht.

Flynn, D., Warhurst, R. (1994): An empirical study of the validation process within requirements determination. Information Systems Journal **4**: 185-212

Galliers, R., Swan, J. (2000): There's more to information systems development than structured approaches: information requirements analysis as a socially mediated process. Requirements Engineering **5**(2): 74-82

Goldin, L., Berry, D. (1997): Abstfinder, a prototype natural language Text Abstraction Finder for Use in Requirements Elicitation. Automated Software Engineering **4**: 375-412

Land, F. (1998): A Contingency Based Approach to Requirements Elicitation and Systems Development. Journal of Systems and Software **40**: 3-6

Mantel, S., Meredith, J., Shafer, S., Sutton, M. (2001): Project management in practice. Wiley and Sons

Van Engelen, J., Van der Zwaan, A. (1994): Bedrijfskundige methodologie 2: een technisch-methodologische context. Bedrijfskunde **66** (2): 85-94 (in Dutch)

Wolthuis, M. (1997): Afstudeerscriptie: Boekhouden op basis van Universele Informatiekunde. University of Limburg. (in dutch)